```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract StudentData {
    struct Student {
        uint id;
        string name;
        uint age;
        string course;
    }

    Student[] public students;
    uint private studentCounter;

    event StudentAdded(uint id, string name, uint age, string course);

    constructor() {
        studentCounter = 0;
    }

    function addStudent(string memory _name, uint _age, string memory _course) public {
        studentCounter++;
        students.push(Student(studentCounter, _name, _age, _course));
        emit StudentAdded(studentCounter, _name, _age, _course);
    }

    function getStudent(uint _id) public view returns (string memory, uint, string memory) {
        for (uint i = 0; i < students.length; i++) {
            if (students[i].id == _id) {
                return (students[i].name, students[i].age, students[i].course);
            }
        }
        revert("Student not found");
    }

    fallback() external payable {
        revert("Fallback triggered: invalid function call");
    }

    function getStudentCount() public view returns (uint) {
        return students.length;
    }
}
```